A person wearing a mask

Description automatically generated with low confidence

**Le logiciel « RecettesCVM »**

**Version : 1.0**

## Le logiciel « RecettesCVM »

Le système « RecettesCVM» permet de gérer des recettes et de faire des propositions de recettes similaires.

Les fonctionnalités principales du système sont :

* Ajouter, modifier, supprimer des recettes
* Faire des recherches dans les recettes présentes dans le système
* Donner le nom de recettes similaires à une recette en cours de modification
* Générer une nouvelle recette en se basant sur les données du système
* Donner des informations générales sur les données du système

A person wearing sunglasses and holding his hand up

Description automatically generated with medium confidence

## Votre mandat

Le logiciel est pratiquement terminé. L’interface graphique ainsi que la logique d’affaires sont déjà programmées, il ne manque que l’implémentation de la persistance des données.

Votre travail consiste donc à terminer le logiciel en complétant la classe RecipeDAO. La JavaDoc décrit ce qui doit être fait pour chaque méthode.

# Informations supplémentaires

* L’application doit être codée à partir de l’application/squelette fourni.
* L’application doit utiliser 2 bases de données parmi les suivantes :
  + BerkeleyDB (pour les photos) + Neo4j, ou
  + BerkeleyDB (pour les photos) + MongoDB
* Le logiciel doit fonctionner sur le poste de travail du professeur sans installation de librairies supplémentaires
* Le projet doit être fait en équipe de 2 au maximum
* Chaque jour de retard entraîne 10% de pénalité sur la note de ce travail
* Si vous utilisez du code pris sur Internet, veuillez citer la source en commentaire, autrement cela pourrait être considéré comme du plagiat
* Il est conseillé d’utiliser Git pour faire la gestion du code source.
* Même s’il n’y a pas beaucoup de données dans vos bases de données, créez les index aux bons endroits. Vous n’avez pas à créer les index en Java, seulement dans les consoles des bases de données (Compass ou neo4j browser), puis indiquer quels sont les index dans votre README.
* Programmez clairement (pas de code inutile, en commentaire ou mal indenté).

### README.md

Un fichier README.md doit être remis avec votre projet. Celui-ci doit contenir :

1- Le nom des coéquipiers

2- Quelle seront les collections (MongoDB) ou libellés (Neo4j) utilisés?

4- Où sont les index, ordre (asc/desc) + type (unique ou pas) ?

### Squelette du programme

Une base du programme a déjà été conçue. Celle-ci intègre déjà l’interface graphique et possède les librairies pour communiquer avec BerkeleyDB (*embedded*), Neo4j et MongoDB.

## Échéancier et livrables

Le projet (le code de l’application complète) doit être remis sur le Z:/Remise. Le logiciel et ses fonctionnalités doivent être présentés à l’enseignant sur son poste de travail lors du dernier cours de la session. Tous comme les mini-projets fait au cours de la session, le module de correction doit être exécuté au moment de la remise.

## Grille de correction générale

|  |  |
| --- | --- |
|  |  |
| README.txt   * Noms des coéquipiers * Nom des collections/libellés utilisés * Emplacement des index, leur type et l’ordre (s’il y a lieu) | % 5 |
| Sauvegarde de recettes | % 15 |
| Liste des recettes | % 15 |
| Suppression | % 5 |
| Suppression totale | % 5 |
| Nombre moyen d’ingrédients par recette | % 5 |
| Nombre de photos dans le système | % 5 |
| Nombre de recettes dans le système | % 5 |
| Le temps maximum d’une recette | % 5 |
| Les recettes similaires | % 10 |
| La dernière recette ajoutée | % 5 |
| Générateur de recettes | % 10 |
| Divers (qualité générale du code) | % 10 |
| Total | **% 100** |